

# Quaternion softmax classifier

R. Zeng, J.S. Wu, Z.H. Shao, L. Senhadji, H.Z. Shu

For the feature extraction of RGB colour images, researchers usually deal with R, G and B channels separately to get three feature vectors, and then combine them together to obtain a long real feature vector. This approach does not exploit the relationships between the three channels of colour images. Recently, attentions have been paid to quaternion features, which take the relationships between channels into consideration and seem to be more suitable for representing colour images. However, there are only few quaternion classifiers for dealing with quaternion features. To meet this requirement, in this paper, we propose a new quaternion classifier, namely, quaternion softmax classifier, which is an extended version of conventional softmax classifier generally defined in complex (or real) domain. We apply the proposed quaternion softmax classifier to two of the most common quaternion features, that is, quaternion principle components analysis (QPCA) feature and colour image pixel feature. The experimental results show that the proposed method performs better than quaternion back propagation neural network (QBPNN) in terms of both accuracy and convergence rate.

**Introduction:** In recent years, research in the field of machine learning was stimulated by unsupervised feature learning approaches, especially deep learning [1]. Features that are obtained by means of learning algorithm have been widely used in computer vision, language detection, voice recognition and so on. How to classify these features effectively is a challenge.

On the other hand, the theory and application of quaternion or hypercomplex algebra, invented by Hamilton [2], has received many attentions in recent years [3-5]. Subsequently, Some researchers did studies in quaternion features extraction, for example, Ell and Sangwine [3, 4] used the quaternion Fourier transform to colour image processing, in which they represent colour image pixels using quaternion (i.e. quaternion pixels). Bihan and Sangwine [5] proposed quaternion principal component analysis (QPCA) as a new feature extraction method for colour image. Risojevic and Babic [6] combined QPCA features with k-means clustering in real domain for colour image classification, they concluded that QPCA is superior to RGB PCA in colour image features extraction. Other researchers did studies in quaternion features classifiers. Arena et al. [7] proposed quaternion back propagation neural network (QBPNN), which is probably the most frequently used classifier for quaternion features and performs better than classical real neural network (NN) when dealing with features classification problem. QBPNN also performs better than classical real neural network in chaotic time series prediction [8]. Chen et al. [9] represented colour face images by quaternion Zernike moment invariants features, which are then classified by QBPNN.

Quaternion features are superior to real features in representing the colour images, however, the frequently used QBPNN classifier still encounter the problem of low classification accuracy and slow convergence rate. Therefore, quaternion features classifiers with high classification accuracy and fast convergence rate are highly needed. To meet this requirement, in this paper, we propose a new quaternion classifier, namely, quaternion softmax classifier, which is an extended version of the conventional softmax classifier generally defined in real domain.

**Quaternion algebra:** A quaternion  $x$  is a hypercomplex number, which consists of one real part and three imaginary parts as

$$x = x_{(s)} + x_{(i)}i + x_{(j)}j + x_{(k)}k \in \mathbb{Q} \quad (1)$$

Where  $\mathbb{Q}$  denotes quaternion number field,  $x_{(s)}, x_{(i)}, x_{(j)}, x_{(k)} \in \mathbb{R}$ ,  $\mathbb{R}$  denotes the real number field, and  $i, j$  and  $k$  are three imaginary units obeying the following rules:  $i^2 = j^2 = k^2 = ijk = -1$ ,  $ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j$ . It should be noted from this that multiplication of quaternion numbers is not commutative. The  $l_2$  norm of quaternion  $x$  is defined as  $\|x\| = \sqrt{x_{(s)}^2 + x_{(i)}^2 + x_{(j)}^2 + x_{(k)}^2}$ . The inner product of two quaternions are  $(x, y) = (x_{(s)}y_{(s)}, x_{(i)}y_{(i)}, x_{(j)}y_{(j)}, x_{(k)}y_{(k)})$ .

**Proposed method:** Given a label set  $Y = \{1, 2, \dots, r\}$  and a quaternion input pattern  $x = (x_1, x_2, \dots, x_n) \in \mathbb{Q}^n$ , the hypothesis to estimate the probability of each label, can be defined as follows:

$$H_\theta(x) = \begin{bmatrix} p(Y=1|\theta_1) \\ \vdots \\ p(Y=r|\theta_r) \end{bmatrix} = \frac{1}{\sum_{h=1}^r \|e^{\theta_h^T x}\|^2} \begin{bmatrix} e^{\theta_1^T x} \\ \vdots \\ e^{\theta_r^T x} \end{bmatrix} \quad (2)$$

Where  $\theta_1, \theta_2, \dots, \theta_r \in \mathbb{Q}^n$ ,  $\theta_h = \{\theta_{h1}, \theta_{h2}, \dots, \theta_{hn}\}$ , are the quaternion parameters of this model, and  $e^{\theta_h^T x} = e^{\theta_{h(s)}x_{(s)}} + e^{\theta_{h(i)}x_{(i)}}i + e^{\theta_{h(j)}x_{(j)}}j + e^{\theta_{h(k)}x_{(k)}}k$ . The summation term normalizes the distribution of quaternion data, makes its summation to one, to meet the probability criteria.

The cost function of quaternion dataset  $X = \{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ , which contains  $m$  patterns, is defined as:

$$J(\theta) = -\frac{1}{m} \sum_{g=1}^m \sum_{h=1}^r \mathbf{1}\{Y^{(g)} = h\} \log \frac{\|e^{\theta_h^T x^{(g)}}\|^2}{\sum_{l=1}^r \|e^{\theta_l^T x^{(g)}}\|^2} \quad (3)$$

Where  $\mathbf{1}$  is a logical value function, that is, when a true statement is given,  $\mathbf{1}\{\cdot\} = 1$ , otherwise  $\mathbf{1}\{\cdot\} = 0$ .

The updates of the quaternion parameters are determined by the gradients of  $J(\theta)$  with respect to the network parameters, then we get  $\theta^{\text{new}} = \theta^{\text{old}} - \nabla \theta$ , where  $\nabla \theta = \partial J(\theta) / \partial \theta_{hq}$ . The quaternion parameters  $\theta_{hq}$  are updated according to the following equations by gradient descend method:

$$\nabla_{\theta_{hq(t)}} J(\theta) = \frac{1}{m} \sum_{g=1}^m \mathbf{1}\{Y^{(g)} = h\} 2 \frac{(e^{\theta_{h(i)}x_{(i)}})^2 x_{q(t)}^{(g)}}{\|e^{\theta_h^T x}\|^2} - 2 \frac{(e^{\theta_{h(i)}x_{(i)}})^2 x_{q(t)}^{(g)}}{\sum_{l=1}^r \|e^{\theta_l^T x}\|^2}, t = s, i, j, k \quad (4)$$

**Results:** Two kinds of quaternion features, quaternion pixels features and QPCA features are used to test the capabilities of our proposed quaternion softmax classifier. A set of experiments was performed on the Georgia Tech Face Database where 450 colour face images composed of 30 individuals with 15 images per individual were used. All images were linearly scaled to 40 by 32 pixels. Some examples are shown in Fig. 1. For each individual, the first 10 images were selected as the training set while the rest as the testing set. For comparison purpose, the recognition rate was also tested by the QBPNN method.



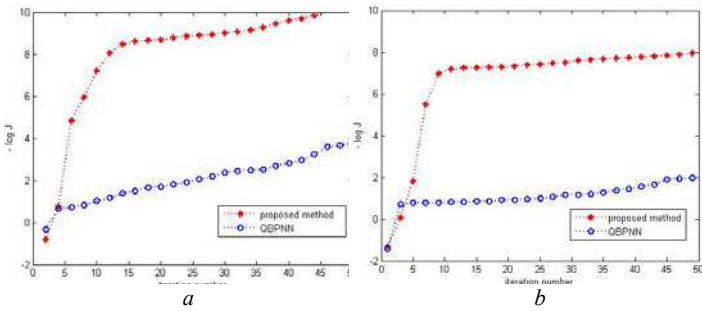
**Fig. 1** Nine images of the Georgia Tech Face Database

To compare with the proposed method effectively, the appropriate selection of the parameters used by the QBPNN is required. Most classification problem used 3-layers neural network. The number of input layer units equals to the number of features. The number of output units equals to the number of classes. Assume that  $N_0, N_1, N_2$  represent the numbers of input layers, hidden layers, output layers, respectively. The hidden size is  $\sqrt{N_0 N_2}$ . The initial weights and bias are set randomly on  $[-u, u]$ , where  $u = \sqrt{6 / (N_0 + N_1 + 1)}$ . If one of the node values of output layers is  $1+i+j+k$  (in practice we use the value that is closest to  $1+i+j+k$  in terms of Euclidean distance), we suppose the test image belongs to this class. The parameters of proposed method are set randomly on  $[-0.005, 0.005]$  for every parts of quaternion.

In learning phase of QBPNN and proposed method, we used min-Func matlab toolbox [10] for numerical optimization. We set optimized method to 'L-BFGS' for both experiments to search the global minimum. In order to obtain enough accuracy, we set the maximum number of iterations to 300. Learning rate and iteration step are set to default values of toolbox.

To clearly illustrate the simulation results, we used the negative logarithm of the cost function ( $-\log J$ ) to describe the evolution of the error with respect to the iteration number.

In quaternion pixels features experiments, every training pattern is normalized by subtracting the mean and dividing by the standard deviation of its elements. For nature images, this corresponds to local brightness and contrast normalization. After normalizing each training picture, the entire training database is processed with a popular method called whitening, which makes the input less redundant (i.e. uncorrelated). Then these processed colour images are represented as a purely imaginary quaternion of the form  $R-i+G-j+B-k$ . So we get desired quaternion pixels features to train the classifier. The performance of the proposed algorithm (95.33%) is better than classical QBPNN (92%) for the quaternion pixel features of the noiseless images. As shown in Fig. 1a, our approach exhibits faster convergence rate than QBPNN method. To consider real situation, two kinds of noises (Gaussian and Salt & Pepper) are added to the raw images in the testing process, before using the above method to pre-processing. The standard deviation (STD) of Gaussian noise varies from 5 to 20. The density (D) of Salt & Pepper is changed from 0.02 to 0.05. All experimental results are shown in Table 1.



**Fig. 1** The convergence rate using two different methods in QPCA features and colour image pixel features respectively.  
a The convergence rate of noiseless colour image pixel features  
b The convergence rate of 30 QPCA features

**Table 1:** Comparison of classification accuracy performed on the GT colour face database using different methods.

Noise type		Proposed method (%)	QBPNN (%)
Gaussian noise (STD)	5	95.33	91.33
	10	92.67	92.67
	15	91.33	91.33
	20	88	87.33
Salt & Pepper (D)	0.02	89.33	88.67
	0.03	86	86
	0.04	84.67	79
	0.05	82	70

**Table 2:** Classification accuracy on QPCA of various feature numbers.

QPCA features (n)	Proposed method (%)	QBPNN (%)
15	94	94
20	96	92.67
25	98	94
30	98	96

In QPCA features of colour images experiments, the entire database is computed with QPCA [5] to get quaternion features. After processing, we have varying quaternion feature for each picture according to the acquirements. Subsequently, every quaternion features of image are put into model in the same manner as above method to test the performance of both classifiers. Both of the proposed method and QBPNN can fit data very well, but the proposed method still performs slightly better. The results are shown in Table 2. The comparisons of two methods in terms of convergence rate are given in Fig. 1b.

**Conclusion:** In this paper, we have proposed a new algorithm for classifying quaternion features, namely, quaternion softmax classifier. The results demonstrate that the proposed method outperforms the conventional QBPNN method in quaternion features classification. In deep learning algorithms, autoencoders in real domain, which combines NN and softmax, have achieved good results in classification. The proposed method also has the capacity to propagate back error to QBPNN. It is possible that autoencoders in quaternion field have advantage over conventional one. This is the basis of deep learning in quaternion data filed. In future, we will focus on deep learning of quaternion data.

**Acknowledgements:** This work was supported by the National Basic Research Program of China under Grant 2011CB707904, by the NSFC under Grants 61201344, 61271312, 11301074, and by the SRFDP under Grants 20110092110023 and 20120092120036, the Project-sponsored by SRF for ROCS, SEM, and by Natural Science Foundation of Jiangsu Province under Grant BK2012329 and by Qing Lan Project.

R. Zeng, J.S. Wu, Z.H. Shao and H.Z. Shu (LIST, the Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China; Centre de Recherche en Information Biomédicale Sino-français, Nanjing 210096, China)  
E-mail: jswu@seu.edu.cn

L. Senhadji (INSERM, U 1099, Rennes 35000, France; LTSI, Université de Rennes 1, Rennes 35000, France; Centre de Recherche en Information Biomédicale Sino-français, Rennes 35000, France)

## References

- Charalampous, K., Kostavelis, I., Amanatiadis, A., and Gasteratos, A.: 'Sparse Deep-Learning Algorithm for Recognition and Categorisation', *Electron. Lett.*, 2012, **48**, (20), pp. 1265-1266, doi: 10.1049/el.2012.1033
- Hamilton, W.R., Hamilton, W.E.: 'Elements of quaternions' (Longmans, Green, & Company, London, 1866)
- Ell, T.A., and Sangwine, S.J.: 'Hypercomplex Fourier Transforms of Color Images', *IEEE Trans. Image Process.*, 2007, **16**, (1), pp. 22-35, doi: 10.1109/TIP.2006.884955
- Sangwine, S.J.: 'Fourier Transforms of Colour Images Using Quaternion or Hypercomplex Numbers', *Electron. Lett.*, 1996, **32**, (21), pp. 1979-1980, doi: 10.1049/el:19961331
- Le Bihan, N., and Sangwine, S.J.: 'Quaternion Principal Component Analysis of Color Images', Conf. ICIP, Barcelona, Catalonia, September 2003, **1**, pp. 809-812, doi: 10.1109/ICIP.2003.1247085
- Risojevic, V., and Babic, Z.: 'Unsupervised Learning of Quaternion Features for Image Classification', Conf. TELSIKS, Serbia, Niš, October 2013, **1**, pp. 345-348, doi: 10.1109/TELSKS.2013.6704945
- Arena, P., Fortuna, L., Muscato, G., and Xibilia, M.G.: 'Multilayer Perceptrons to Approximate Quaternion Valued Functions', *Neur. Netw.*, 1997, **10**, (2), pp. 335-342, doi: 10.1016/S0893-6080(96)00048-2
- Arena, P., Baglio, S., Fortuna, L., and Xibilia, M.G.: 'Chaotic time series prediction via quaternionic multilayer perceptrons', Conf. IEEE Conf. Syst. Man. Cyber., Vancouver, BC, Canada, October 1995, **2**, pp. 1790-1794, doi: 10.1109/ICSMC.1995.538035
- Chen, B.J., Shu, H.Z., Chen, G., and Ge, J.: 'Color Face Recognition Based on Quaternion Zernike Moment Invariants and Quaternion Bp Neural Network', *Applied Mech. Mater.*, 2014, **446**, pp. 1034-1039, doi: 10.4028/www.scientific.net/AMM.446-447.1034
- Schmidt, M.: 'minFunc: Matlab software for unconstrained differentiable multivariate optimization', <http://www.di.ens.fr/~mschmidt/Software/minFunc.html>, accessed June 2014